Information Sciences 316 (2015) 550–566

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Patterns from nature: Distributed greedy colouring with simple messages and minimal graph knowledge

Lei Xu [a,b], Peter Jeavons [b,*]

[a] Audaque Data Technology Ltd., Software Building, 9 Gaoxin Middle First Road, Nanshan District, Shenzhen 518000, China
[b] Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, UK

A R T I C L E I N F O

Article history:
Received 10 September 2013
Received in revised form 21 February 2014
Accepted 26 June 2014
Available online 10 July 2014

Keywords:
Bio-inspired algorithm
Distributed computing
Greedy colouring
Randomised algorithm

A B S T R A C T

A well-established problem in global optimization is the problem of colouring the vertices of an arbitrary graph using the minimal number of colours, such that adjacent vertices are assigned different colours. One way to restrict the number of colours used is to allow only greedy colourings. A greedy colouring is an assignment of colours to the vertices of a graph that can be obtained by an algorithm that considers each vertex in turn and assigns the first colour that is not already assigned to some neighbour. An optimal colouring can always be obtained in this way, by choosing an appropriate order on the vertices.

Recently, a new bio-inspired approach to distributed pattern formation has been proposed, based on modelling the neurological development of the fruit fly. Building on that approach, we propose a new simple randomised algorithm for distributed greedy colouring using only local processing at the vertices and messages along the edges. In our approach the processors exchange only simple messages representing potential colour values and each processor has minimal graph knowledge. We discuss two variations of this algorithm, and investigate their time complexity and message complexity both theoretically and experimentally.

In addition, we show experimentally that the number of colours used turns out to be optimal or near-optimal for many standard graph colouring benchmarks. Thus, for distributed networks, our algorithm serves as an effective heuristic approach to computing a colouring with a small number of colours.

1. Introduction

One of the most fundamental graph-algorithmic problems is to colour the vertices of a graph so that adjacent vertices are assigned different colours. This problem is known as the *graph colouring* problem [19].

To reduce the overall number of different colours used, it is often useful to assume that there is a total ordering on the colours and impose the further restriction that there is no vertex whose colour assignment can be changed to a colour that is lower in the ordering without clashing with at least one of its neighbours. A graph colouring with this additional restriction is called a *Grundy colouring* [14,10].

It is trivial to obtain a Grundy colouring using an algorithm with a central control: simply consider the vertices of the graph in some arbitrary order and assign each vertex its first available colour. This approach is known as *greedy colouring*, so the problem of finding a Grundy colouring is usually referred to as the greedy colouring (GC) problem.

* Corresponding author.
E-mail addresses: lei.xu@cs.ox.ac.uk (L. Xu), peter.jeavons@cs.ox.ac.uk (P. Jeavons).

http://dx.doi.org/10.1016/j.ins.2014.06.035

To obtain a Grundy colouring efficiently without any central control, using only local processing at each vertex and messages along the edges, is a much more challenging task, and is an important problem in distributed computing [10]. Distributed graph colouring serves as a basic building block in many other distributed algorithms, and has applications in networking, wireless sensor networks and in particular in frequency assignment in radio-communication networks [23,27,29,24].

Recently, a new algorithmic approach to another well-established problem in distributed computing was inspired by the study of the development of the nervous system of the fruit fly [2,1]. The problem that was considered is known as the *maximal independent set* (MIS) selection problem. In this problem the task is to elect a set of local leaders in a network of connected processors such that every processor is either a leader or connected to a leader, and no two leaders are connected to each other. During the development of the nervous system of the fruit fly, certain cells in the pre-neural clusters of the fly specialise to become sensory organ precursor (SOP) cells. The SOP pattern formation is very similar to the MIS selection: each cell either becomes an SOP or a neighbour of an SOP, and no two SOPs are neighbours. The similarity of SOP cell selection and MIS selection in a graph is illustrated in Fig. 1.

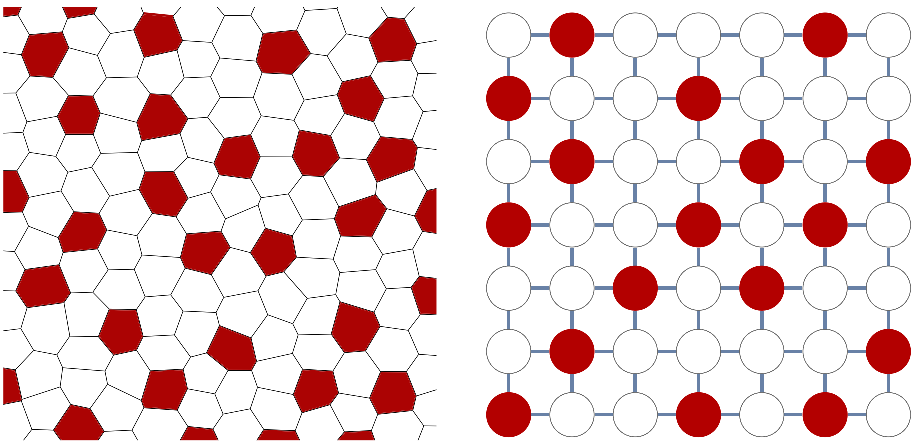
The cells of the fruit fly appear to form the SOP pattern without clear central control using only simple local interactions between certain membrane-bound proteins [3,6]. These interactions set up a bistable switch at each cell between two states, one of which is distinguished. A key feature of the biological system is that the probability that an isolated cell will enter the distinguished state increases over time, but there is a strong lateral inhibition mechanism to ensure that neighbouring cells cannot both occupy the distinguished state. By modelling this probabilistic switching mechanism, a new approach to the MIS selection problem was proposed in [2] and later improved in [1]. The computational problem of distributively selecting an MIS can therefore now be tackled in a novel way by learning from the neurological development of the fruit fly in the natural world. This new approach requires very little knowledge of the environment at each processor, and requires only very simple (one-bit) messages that are broadcast to neighbouring processors. The probability of sending such a message varies over time, and any cell broadcasting a message and not receiving a message in return permanently enters the distinguished state.

In this paper, we extend this bio-inspired MIS selection approach to solve the distributed greedy colouring problem with a very simple mechanism. More specifically, we consider an extremely harsh distributed computing model for greedy colouring: each vertex in the graph hosts a processor. The communication is synchronous and proceeds in discrete time steps. The processors have minimal knowledge about the graph, and the only messages they can send at each time step are single colour values, which are broadcast to all of their neighbours.

We show that even under such harsh conditions we can construct a randomised algorithmic scheme that computes a Grundy colouring distributively. We consider two different variants of this algorithmic scheme. In the first version, Algorithm 1, the processors at each vertex of the graph have no information at all about the graph, and use a predetermined sequence of probability values to decide whether or not to send a message at each step. This version computes a Grundy colouring after an expected number of steps which is $O(\Delta^2 \log^2 n)$, where n is the number of vertices and Δ is the maximum degree of the graph. The expected number of messages sent by each node is $O(\Delta \log n)$.

In the second version, Algorithm 2, the processors at each vertex have information about the total number n of the vertices of the graph, and use this information to select a tailored sequence of probability values. This version computes a Grundy colouring after an expected number of steps which is $O(\Delta \log^2 n)$. The expected number of messages sent by each node is again $O(\Delta \log n)$.

Our two new algorithms are compared with previous distributed greedy colouring algorithms in Table 1, where Δ, m, n denote respectively the maximum degree, the number of edges, and the number of vertices of the graph.



(a) The selection of SOP cells in the fruit fly (b) An MIS selected from a 7 × 7 grid graph

Fig. 1. The similarity between SOP cell selection in the fruit fly and MIS selection in a graph.